

**CERTIFIED COPY OF
PRIORITY DOCUMENT**



**Prioritätsbescheinigung über die Einreichung
einer Patentanmeldung**

Aktenzeichen: 100 55 169.6

Anmeldetag: 08. November 2000

Anmelder/Inhaber: Siemens Aktiengesellschaft;
München/DE

Bezeichnung: Industrielle Steuerung auf der Basis Technologischer
Objekte

Priorität: 03.08.2000 DE 100 37 971.0
03.08.2000 DE 100 37 990.7

IPC: G 05 B 19/04

**Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ur-
sprünglichen Unterlagen dieser Patentanmeldung.**

München, den 20. Dezember 2000
Deutsches Patent- und Markenamt

Der Präsident
Im Auftrag

Aquarius

Beschreibung

Industrielle Steuerung auf der Basis Technologischer Objekte

5 Die Erfindung bezieht sich auf eine industrielle Steuerung für technische Prozesse, insbesondere für Produktionsmaschinen.

10 Ferner bezieht sich die Erfindung auf ein Verfahren zur Erstellung einer industriellen Steuerung, insbesondere für ein bestimmtes Einsatzgebiet, z.B. Einsatz bei Verpackungsmaschinen.

15 Eine industrielle Steuerung kann dabei ein eigenes Gerät sein, sie kann aber auch in einen Computer, einem PC, einem eigenständigen Gerät oder einem Antrieb integriert sein.

20 Bisher bekannte industrielle Steuerungen zur Automatisierung technischer Prozesse basieren im Wesentlichen entweder auf einer "SPS-Funktionalität", einer "MC-Funktionalität" oder auf einer Technologiefunktionalität. Da im Rahmen solcher Funktionalitäten ein gewisser Funktionsumfang fest vorge-

25 schrieben ist, ist eine optimale Anpassung an die Anforderungen eines speziellen Prozesses häufig nur bedingt möglich, wobei im konkreten Anwendungsfall oft eine ganze Gruppe von Funktionen überflüssig ist (z.B. ist beim Einsatz einer MC-Steuerung für Werkzeugmaschinen, evtl. vorhandene Funktionalität für Verpackungsmaschinen überflüssig).

30 Aus der DE 197 40 550 ist außerdem eine Vorrichtung zum Steuern eines technischen Prozesses und/oder zur Steuerung der Bewegung einer Verarbeitungsmaschine bekannt, die ein Steuerprogramm abarbeitet. Dieses Steuerprogramm besteht aus einer Vielzahl von Softwaremodulen. Prozesssteuerungsfunktionalitäten von an sich bekannten speicherprogrammierbaren Steuerungen

35 und Bewegungsfunktionalitäten von an sich bekannten MC-Steuerungen sind in einem einheitlichen, konfigurierbaren

Steuerungssystem verwirklicht. Die einzelnen Software-Module werden hier jedoch durch jeweils eine Teilsteuerung abgearbeitet, so dass für jedes Software-Modul eine zentrale Recheneinheit vorzusehen ist.

5

Weiterhin ist aus der DE 198 53 205 ein Verfahren zur Steuerung technischer Prozesse bekannt, das auf einer Instanziierbarkeit sowie einer bedarfsgerechten Verschaltung von Software-Komponenten mit vorgebbarer, zumindest parametrierbarer Funktionalität basiert. Die Verschaltung und die Projektierung der Software-Komponenten erfolgt hierbei allerdings noch nicht optimal.

Der Erfindung liegt daher die Aufgabe zugrunde, für jeweils unterschiedliche Steuerungsaufgaben und unterschiedliche Randbedingungen bzw. Anforderungen des zugrunde liegenden technischen Prozesses in einfacher Weise optimale Ausprägungen einer industriellen Steuerung sowohl hinsichtlich ihrer Steuerungsstruktur als auch hinsichtlich ihrer Funktionalität zu erstellen.

Die Erfinder sind dabei von der Erkenntnis ausgegangen, dass das Runtime- und/oder Engineering-System der industriellen Steuerung sowohl SPS- als auch Bewegungs- und/oder Technologie-Funktionalität bedient und dass durch die Möglichkeit des dynamischen Zuladens von Funktionscode in das Runtime- und/oder Engineering-System der industriellen Steuerung jeweils eine optimale Ausprägung, d.h. Skalierung der Steuerung möglich sein müsste.

30

Gemäß der Erfindung wird die oben genannte Aufgabe für eine industrielle Steuerung der eingangs genannten Art dadurch gelöst, dass

35 a) die Steuerung ein allgemein einsetzbares, vorzugsweise technologieneutrales, Basissystem für die Steuerungsgrundfunktionalität besitzt,

- b) die technologische Funktionalität der Steuerung über technologische Objekttypen bereitgestellt ist,
- c) eine technologische Skalierung hinsichtlich der Funktionalität der Steuerung durch zuladbare Technologische Objekttypen erfolgt und
- d) eine auf die jeweilige Applikation vom Anwender zuschneidbare Auswahl und Instanziierung der Objekttypen erfolgt.

Ein Technologisches Objekt repräsentiert vorzugsweise eine Komponente der realen Welt. Im Kontext industrieller Steuerungen können dies z.B. Komponenten von Werkzeugmaschinen oder Produktionsmaschinen sein. Die Technologischen Objekte stellen eine definierte technologische, abgeschlossene Funktionalität bereit. Sie können untereinander verschaltet werden, um komplexe technologische Aufgaben zu realisieren. Dadurch dass die technologische Funktionalität der Steuerung durch Technologische Objekte, die vorzugsweise reale Komponenten, gebildet wird, ist einem Anwender oder Nutzer der Steuerung die technologische Mächtigkeit, d.h. die Fähigkeit der Steuerung sofort transparent. Als softwaretechnologische Einheit kann ein Technologisches Objekt außerdem von einem Anwender sehr leicht in unterschiedlichen Applikationen und Steuerungen wiederverwendet werden. Ein Anwender kann bei der Nutzung von Technologischen Objekten von deren Implementierung abstrahieren. Vom Anwender in seinen Anwenderprogrammen direkt einsetzbare Technologische Objekte entstehen durch ihre Instanziierung aus Technologieobjekttypen. Aus einem einmal definierten Technologieobjekttyp können beliebig viele Instanzen von Technologischen Objekten gewonnen werden. Dadurch dass die Instanziierung sowohl im Engineering-System als auch im Runtime-System erfolgen kann, ist es für einen Anwender sehr leicht und sehr komfortabel möglich, die Technologischen Objekte in seinen Anwendungen zu verwenden. Die funktionale Mächtigkeit einer Steuerung ist somit sehr leicht erweiterbar. Die Erweiterbarkeit wird lediglich durch HW-Restriktionen (z.B. CPU-Leistung oder Speicherbeschränkungen) begrenzt.

Weiterhin hat der Anwender die Möglichkeit das vorhandene Basissystem für die Steuerungsgrundfunktionalität um solche Funktionalitäten zu erweitern, die er wirklich für seine Anwendungen benötigt. Dies geschieht dadurch, dass er explizit
5 bestimmte benötigte Technologische Objekte zum Basissystem der Steuerung hinzulädt. Ein Anwender kann sich somit individuell eine Steuerung mit einer bestimmten Funktionalität beschaffen. Üblicherweise in Steuerungen vorhandene nichtbenötigte Funktionalitäten werden dadurch vermieden und verursachen somit keinen Overhead.
10

Ein weiterer Vorteil liegt in der Entwicklung und in der Produktion solcher skalierbaren Steuerungen. Steuerungen, die mit einer notwendigen Grundfunktionalität (Basissystem) ausgeliefert werden, lassen sich in großer Stückzahl sehr einfach herstellen (economies of scale).
15

Eine erste vorteilhafte Ausgestaltung der Erfindung liegt darin, dass das Laden und die Verteilung der Technologischen Objekte auf unterschiedlich oder gleich performante Hardware-Systeme und/oder Laufzeitsysteme erfolgt. Technologische Objekte sind plattform- bzw. hardware-unabhängig. Sie enthalten keine plattform- bzw. hardware-spezifischen Eigenschaften und können somit sehr leicht auf unterschiedliche Hardware-Systeme und/oder Laufzeitsysteme geladen und verteilt werden.
20 Durch die Möglichkeit der Lad- und Verteilbarkeit der Technologischen Objekte auf unterschiedlich oder gleich performante Hardware-Systeme kann ein Anwender Technologische Objekte sehr flexibel benutzen und einsetzen. Er muss sich bei der Ladbarkeit und der Verteilbarkeit der Technologischen Objekte nicht um Restriktionen bezüglich der zugrundeliegenden Performance der Hardware- und/oder von Laufzeit-Systemen kümmern.
25
30

35 Eine weitere vorteilhafte Ausgestaltung der zugrunde liegenden Erfindung liegt darin, dass das Laden und die Verteilung der Technologischen Objekte auf unterschiedlich oder gleich

performante Hardware- und/oder Laufzeit-Systeme innerhalb eines Projektes erfolgt, wobei sich ein Projekt auf Daten und/oder Programme von einer oder mehreren Steuerungseinheiten bezieht. Ein Anwender hat somit die Möglichkeit innerhalb eines Projektes Geräte unterschiedlicher Hardware einzusetzen, die auch unterschiedlich performant sein können, auf dieer Technologische Objekte leicht und flexibel verteilen kann, ohne die unterschiedliche Performance der Geräte berücksichtigen zu müssen.

10

Eine weitere vorteilhafte Ausgestaltung der zugrunde liegenden Erfindung liegt darin, dass die Verteilung der Funktionalität der Technologischen Objekte auf miteinander in Echtzeit durch taktsynchron äquidistant kommunizierende Steuerungseinheiten erfolgt. Die Technologischen Objekte können somit auf Geräte bzw. Steuerungseinheiten verteilt werden, die über ein Kommunikationsmedium in Verbindung stehen, das eine taktsynchrone äquidistante Kommunikation erlaubt. Somit können die Technologischen Objekte in Echtzeit miteinander kommunizieren. In einem Projekt sind die Instanzen von Technologischen Objekttypen eindeutig referenzierbar und können (HW-) plattformübergreifend genutzt werden.

15

20

25

30

Eine weitere vorteilhafte Ausgestaltung der vorliegenden Erfindung liegt darin, dass die Funktionalität der Steuerung durch die Zuladbarkeit beliebiger Technologischer Objekte flexibel erweiterbar ist. Damit hat der Anwender die Möglichkeit eine funktionale Skalierung seiner Steuerung zu erreichen. Er kann somit sehr einfach die Funktionalität der Steuerung an die zugrundeliegenden und vorliegenden Bedürfnisse und Randbedingungen anpassen. Die Erweiterbarkeit bezieht sich sowohl auf Gerätefunktionalität als auch auf technologische Funktionalität.

35

Eine weitere vorteilhafte Ausgestaltung der Erfindung liegt darin, dass die Technologischen Objekte Programmierinterfaces und/oder Systemvariablen und/oder Alarmer enthalten. Durch die

Programmierinterfaces können die Technologischen Objekte in einem Anwenderprogramm aufgerufen und verwendet werden. Über die Systemvariablen werden zum einen die Zustandsdaten der Technologischen Objekte dem Anwenderprogramm bereitgestellt, 5 zum anderen enthalten die Systemvariablen parametrierbare Einstellungen für die Programmierung, die aus dem Anwenderprogramm heraus verändert werden können. Über Strukturen können die Werte von Systemvariablen konsistent ausgelesen werden. Systemvariablen werden beim Zugriff aus dem Anwenderprogramm vom System aktualisiert, soweit sie Status und Zustand 10 wiedergeben. Technologische Alarmer werden am Technologischen Objekt festgestellt und erzeugt. Ihr Verhalten auf die Programmbearbeitung (globale Reaktion) ist für jeden Fehler instanzspezifisch am Engineering-System bei der Inbetriebnahme einstellbar. Die Alarmer besitzen zusammen mit der TO-Instanz 15 einen eindeutigen Identifikator.

Eine weitere vorteilhafte Ausgestaltung der vorliegenden Erfindung liegt darin, dass die Technologischen Objekte in eine 20 Programmierungsumgebung integriert sind. Dadurch ist es einem Anwender sehr leicht und komfortabel möglich, die Technologischen Objekte in seinen Anwenderprogrammen zu verwenden.

Eine weitere vorteilhafte Ausgestaltung der vorliegenden Erfindung liegt darin, dass eine Standardprogrammierungsumgebung 25 verwendet wird. Standardprogrammierungsumgebungen sind z.B. IEC1131, C++ oder Java. Durch die Einbindung der Technologischen Objekte in IEC1131 hat der Anwender die Möglichkeit in einer für industrielle Belange genormten Programmierungsumgebung 30 direkt Technologische Objekte zu verwenden. Durch die Integration der Technologischen Objekte in C++ oder Java hat der Anwender darüber hinaus die Möglichkeit Technologische Objekte in einer objektorientierten Programmierungsumgebung direkt zu verwenden. Durch die Integration der Technologischen Objekte 35 in die genannten Programmierungsumgebungen wird der Sprachumfang bzw. der Funktionsumfang dieser Programmierungsumgebungen bezüg-

lich ihrer technologischen Funktionalität und auf Belange industrieller Steuerungen hin erweitert.

5 Eine weitere vorteilhafte Ausgestaltung der vorliegenden Erfindung liegt darin, dass eine rückwirkungsfreie Programmierung eines Technologischen Objektes bezüglich der anderen vorhandenen Technologischen Objekte und des Steuerungsbasis-
systems, sofern nicht explizit eine Rückwirkung programmiert bzw. projektiert ist, vorgesehen ist. Der Anwender kann somit
10 das Verhalten eines Technologischen Objektes unabhängig von Rückwirkungen anderer Technologischen Objekte oder des Steuerungsbasisystems programmieren. Wenn erforderlich oder gewünscht kann er aber explizit eine Rückwirkung programmieren bzw. projektieren. Die Flexibilität des Anwenders bei der
15 Programmierung Technologischer Objekte wird dadurch erhöht.

Eine weitere vorteilhafte Ausgestaltung der vorliegenden Erfindung liegt darin, dass die Nutzung der Technologischen Objekte Hardware- bzw. Plattformneutral erfolgt. Der Anwender
20 kann somit bei der Nutzung der Technologischen Objekte vollkommen von der zugrunde liegenden Hardware- bzw. Plattform abstrahieren. Er kann sich bei der Verwendung von Technologischen Objekten in seinen Anwenderprogrammen somit vollkommen auf die technologische Funktionalität dieser Objekte konzentrieren.
25

Eine weitere vorteilhafte Ausgestaltung der vorliegenden Erfindung liegt darin, dass Technologische Objekte bezüglich ihrer Funktionalität zu anderen Technologischen Objekten hierarchische Beziehungen und/oder Datenflussbeziehungen aufweisen. Damit lassen sich Strukturen und Beziehungen von realen Umgebungen direkt auf Technologische Objekte abbilden. So
30 enthält z.B. das Technologische Objekt "Gleichlaufachse" die Funktionalität des Technologischen Objektes "Positionierachse", das wiederum die Funktionalität des Technologischen Objektes "Drehzahlachse" enthält.
35

Eine weitere vorteilhafte Ausgestaltung der vorliegenden Erfindung liegt darin, dass insbesondere für Bewegungssteuerungen Elemente wie Achsen, Geber, Nocken, Messtaster, Gleichlauf und Kurvenscheiben als Technologische Objekte vorhanden sind. Diese Technologischen Objekte repräsentieren typische Elemente aus dem Kontext von Bewegungssteuerungen. Ihre Repräsentation als Technologische Objekte kann ein Anwender direkt in seinen Anwendungen verwenden.

10 Eine weitere vorteilhafte Ausgestaltung der vorliegenden Erfindung liegt darin, dass die Technologieobjekttypen zu Technologiepaketen zusammenfassbar sind. Durch die Clusterung und Zuordnung von Technologieobjekttypen zu Technologiepaketen wird zum einen eine Strukturierung und Klassifizierung erreicht und zum anderen sind die Technologiepakete ein geeignetes Mittel, um die Technologieobjekttypen auf das Runtime-System einer Steuerung zu laden.

20 Eine weitere vorteilhafte Ausgestaltung der vorliegenden Erfindung liegt darin, dass die Programmerstellung vor Zuordnung der Hardware-Plattform erfolgt. Technologische Objekte können somit in Programmen unabhängig von der Ziel-Hardware verwendet werden. Dadurch wird die Wiederverwendung von Software erleichtert und erhöht.

25 Eine weitere vorteilhafte Ausgestaltung der vorliegenden Erfindung liegt darin, dass die Anzahl der Instanziierungen von Technologieobjekttypen flexibel einstellbar ist. Ein Anwender wird somit bei der Anzahl von Instanzen in seinem Anwenderprogramm nur durch Hardware-Restriktionen (z.B. CPU-Leistung oder Speicherplatz) eingeschränkt.

35 Eine weitere vorteilhafte Ausgestaltung der vorliegenden Erfindung liegt darin, dass die Instanzen von Technologieobjekttypen auf einer oder auf mehreren Steuerungseinheiten verteilt abgelegt werden. Dadurch ist es möglich, die Last in einem System flexibel zu verteilen und zu balancieren.

Eine weitere vorteilhafte Ausgestaltung der vorliegenden Erfindung liegt darin, dass die Befehle von Technologischen Objekten synchron und/oder asynchron genutzt werden. Dadurch wird eine zyklische Programmiermöglichkeit im Sinne einer

5 Speicherprogrammierbaren Steuerung (SPS) und eine ablauforientierte Programmiermöglichkeit (wie bei Bewegungssteuerungen üblich) unterstützt. Der Anwender kann dabei angeben, ob ein Befehl synchron oder asynchron ausgeführt werden soll.

10 Gemäß der Erfindung wird die oben genannte Aufgabe für ein Verfahren der eingangs genannten Art durch die folgenden aufeinander folgenden Schritte gelöst:

- 15 a) Verwendung eines Basissystems mit einer vorzugsweise technologieneutralen Grundfunktionalität,
- b) Auswahl und Instanziierung der für die Anwendung relevanten Technologischen Objekttypen und Konfiguration der erhaltenen Technologischen Objekte im Engineering-System,
- 20 c) Hinzuladen der ausgewählten Technologischen Objekttypen ins Runtime-System,
- d) Laden und Nutzung der instanziierten Technologischen Objekte und der Instanziierungsinformation im Runtime-System,
- 25 e) Erstellung der Anwenderprogramme unter Nutzung der instanziierten Technologischen Objekte und
- 30 f) Laden der Anwenderprogramme ins Runtime-System.

Dadurch hat der Anwender die Möglichkeit auf eine systematische und folgerichtige Weise die Funktionalität einer gewünschten Steuerung zu erreichen, wobei sichergestellt ist,

35 dass die erhaltene Steuerung keinen funktionalen Overhead beinhaltet.

Die wesentlichen, mit der Erfindung erzielten Vorteile bestehen also insbesondere darin, dass ein Anwender direkt technologische Funktionalität in seinen Anwendungen verwenden kann, die ihm durch Technologische Objekte, die Elemente der realen
5 Welt entsprechen, in einer für ihn adäquaten Weise verwenden kann.

Ein weiterer Vorteil liegt darin, dass die Funktionalität von industriellen Steuerungen sozusagen "plug and play"-mäßig de-
10 diziert erweiterbar ist. Auf diese Weise wird eine technologische Skalierung der Steuerung erreicht.

Ein Ausführungsbeispiel der Erfindung ist in der Zeichnung dargestellt und wird im folgenden erläutert.
15 Dabei zeigen:

FIG 1 in einem Strukturbild ein Engineering-System, das zugehörige Runtime-System und den zu steuernden technischen Prozess,
20

FIG 2 zeigt in einem Übersichtsbild wie ein Anwenderprogramm auf technologische Funktionalität im Runtime-System zugreift,

25 FIG 3 zeigt in einer abstrakten Schemadarstellung ein Technologisches Objekt mit Anwenderschnittstelle,

FIG 4 zeigt in Form eines sog. Verschaltungsdiagrammes Technologische Objekte, die einen Gleichlaufverbund darstellen,
30

FIG 5 zeigt einen Gleichlaufverbund mit Umschaltmöglichkeiten zwischen verschiedenen Leitwertquellen und Gleichlaufgesetzen, ebenfalls in Form eines Verschaltungsdiagrammes,
35

FIG 6 zeigt in Form eines Verschaltungsdiagrammes die Verschaltung des Technologischen Objektes Messtaster,

5 FIG 7 zeigt in Form eines Verschaltungsdiagrammes die Verschaltung des Technologischen Objektes Nocke,

FIG 8 zeigt in Form eines Verschaltungsdiagrammes Verschaltungen mit Gleichlauftechnologieobjekten,

10 FIG 9 zeigt ebenfalls in Form eines Verschaltungsdiagrammes die Zuordnung eines Technologischen Objektes Kurvenscheibe zum mehreren Gleichlaufobjekten und

15 FIG 10 zeigt in einem Übersichtsbild die Clusterung von Technologieobjekttypen zu einem Technologiepaket.

In der Darstellung gemäß FIG 1 wird in Form eines Strukturbildes gezeigt, dass die Steuerung eines technischen Prozesses P über mindestens ein Runtime-System RTS1-RTS3 von industriellen Steuerungen erfolgt. Die Verbindung zwischen den Runtime-Systemen RTS1-RTS3 der Steuerung und dem technischen Prozess P geschieht bidirektional über Ein-/Ausgänge EA1-EA3. Die Programmierung der Steuerung und damit das Festlegen des Verhaltens der Runtime-Systeme RTS1-RTS3 geschieht im Engineering-System ES. Das Engineering-System ES enthält Werkzeuge für die Konfigurierung, Projektierung und Programmierung für Maschinen bzw. für die Steuerung technischer Prozesse. Die im Engineering-System ES erstellten Programme werden über die Informationspfade I1-I3 jeweils in die Runtime-Systeme RTS1-RTS3 der Steuerungen übertragen. Durch die drei Punkte ist angedeutet, dass weitere Steuerungen und Runtime-Systeme vorhanden sein können. Bezüglich seiner Hardware-Ausstattung besteht ein Engineering-System ES üblicherweise aus einem Computersystem mit Graphikbildschirm (z.B. Display), Eingabehilfsmitteln (z.B. Tastatur und Maus), Prozessor, Arbeits- und Sekundärspeicher, eine Einrichtung für die Aufnahme computerlesbarer Medien (z.B. Disketten, CD's) sowie

20
25
30
35

Anschlusseinheiten für einen Datenaustausch mit anderen Systemen (z.B. weiteren Computersystemen, weitere Steuerungen für technische Prozesse) oder Medien (z.B. Internet). Eine Steuerung besteht üblicherweise aus Eingabe- und Ausgabeeinheiten, sowie aus Prozessor und Programmspeicher.

Darstellung gemäß FIG 2 zeigt zwei Runtime-Systeme RTS4 und RTS5 von industriellen Steuerungen, dargestellt als Rechteck. Die Runtime-Systeme RTS4 und RTS5 enthalten jeweils einen UMC-Kernel UMC-K, sowie die Technologischen Objekte TO1 bis TOn, wobei die jeweiligen UMC-Kernels als auch die Technologischen Objekte unterschiedlich sein können, die Technologischen Objekte auch in ihrer Anzahl. Der UMC-Kernel UMC-K stellt das Basissystem der Steuerung dar, dieses Basissystem enthält die Grundfunktionalität der Steuerung. Der UMC-Kernel UMC-K ist in rechtwinkliger Stufenform dargestellt. Zu ihm können Technologische Objekte TO1 bis TOn hinzugeladen werden. Durch dieses Hinzuladen wird der Funktionsumfang des Basissystems erweitert. Die Technologischen Objekte TO1 bis TOn sind als Rechtecke dargestellt, durch ihre Anordnung in FIG 2 wird angedeutet, dass sie den UMC-Kernel UMC-K erweitern. Durch die drei Punkte wird angedeutet, dass ein bis mehrere Technologische Objekte TO1 bis TOn hinzugeladen werden können und somit eine technologische Skalierung der gesamten Steuerung erreicht wird. Zentriert am oberen Rand von FIG 2 ist das Anwenderprogramm AP in Form einer schematischen Papierfahne dargestellt. Durch die Zugriffspfeile ZGP1 bis ZGP4 ist dargestellt, dass ein Anwender in seinem Anwenderprogramm AP direkt auf Funktionalitäten des UMC-Kernels UMC-K aber auch auf Funktionalitäten der Technologischen Objekte TO1 bis TOn zugreifen kann, sowohl von RTS4 als auch von RTS5 oder von einem weiteren Runtime-System (ebenfalls angedeutet durch drei Punkte). Diese angebotene Funktionalitäten der Runtime-Systeme RTS4 und RTS5 (oder von weiteren Runtime-Systemen) kann ein Anwender direkt in seinem Anwendungsprogramm AP verwenden.

Zur Präzisierung: Zur Erweiterung des Basissystems eines Runtime-Systems werden Technologieobjekte üblicherweise in Form von Technologieobjekttypen hinzugeladen. Solche Technologieobjekttypen sind z.B. Achsen, Nocken, Kurvenscheiben oder
5 ähnliches. Technologieobjekttypen sind instanzierbar. Ein Anwender verwendet in seinen Anwendungsprogrammen AP für konkrete Applikationen Instanzen von Technologieobjekttypen. Solche Instanzen sind dann projektweit eindeutig definiert und identifizierbar. Die direkte Verwendung von zugeladenen
10 Technologischen Objekten in Anwenderprogrammen AP als jeweils eigenständige Programmobjekte wäre prinzipiell auch denkbar, ist aber für einen Anwender für die Programmerstellung unflexibel.

15 Darstellung gemäß FIG 3 zeigt in einer abstrakten Schemadarstellung die Anwendersicht eines Technologischen Objektes, d.h. einer Instanz eines Technologieobjekttyps. Diese Spezifikation eines Technologischen Objektes TOS wird als Rechteck dargestellt, das aus fünf Teilen besteht. Der oberste erste
20 Teil, abgetrennt von den folgenden Teilen durch eine durchgezogene Linie, enthält den Typ des zugrunde liegenden Technologischen Objektes (TO-Type) und den TO-Identifizier, d.h. die projekteindeutige Bezeichnung der Instanziierung. Der nächst folgende Teil enthält die Konfigurationsdaten (Configuration
25 Data) mit den Konfigurationsvariablen <configuration variable_1> bis <configuration variable_n>. Über die Konfigurationsdaten wird das Technologische Objekt in seiner grundsätzlichen Wirkungsweise eingestellt. Die Konfigurationsdaten werden über das Engineering-System (ES; FIG 1) eingestellt
30 und können optional über Zugriffsfunktionen aus dem Anwenderprogramm (AP; FIG 2) heraus gelesen bzw. geschrieben werden. In der Darstellung gemäß FIG 3 werden die Konfigurationsdaten durch eine gestrichelte Linie von den Systemvariablen (System
35 Data) abgetrennt. Die Systemvariablen <system variable_1> bis <system variable_m> sind aus dem Anwenderprogramm (AP; FIG 2) heraus veränderbar und wie Programmvariable nutzbar. Systemvariablen können lesbar oder les-/schreibbar sein. Durch die

Systemvariablen werden außerdem die Zustände von Technologischen Objekten repräsentiert. Zustandsübergänge können durch Ereignisse und/oder Befehle ausgelöst werden. Über die Konfigurationsdaten und die Systemvariablen erfolgt die Parametrierung der Technologischen Objekte. Der nächste Abschnitt sind die Befehle (Commands), von den Systemvariablen ebenfalls durch eine gestrichelte Linie getrennt. Die Befehle <command_1> bis <command_xy> stellen aufrufbare Funktionen dar, die die Funktionalität eines Technologischen Objekts repräsentieren. Diese Funktionen haben definierte Bezeichner, Funktionsparameter und lokale Werte. Die Funktionen können Parameter besitzen. Beim Aufruf von Funktionen können optionale Parameter weggelassen werden, hierfür werden dann Defaultwerte eingesetzt. Zusätzlich zur technologischen Funktionalität besitzt ein Technologisches Objekt aber auch Befehle, die das Grundverhalten des Technologischen Objektes bestimmen, z.B.

- Befehl zum Rücksetzen in einem definierten Ausgangszustand
- Befehl um einen anstehenden Fehler gezielt rückzusetzen
- Befehle um in den Simulationsbetrieb zu setzen und rückzusetzen (Im Simulationsbetrieb erfolgt ein Durchlauf des Programms ohne konkrete Ausgabe an die Aktoren, bzw. Einlesen von den Sensoren).
- Befehle um das Technologische Objekt aktiv/inaktiv zu setzen
- Auskunftsfunktionen.

Der nächste Abschnitt der Spezifikation eines Technologischen Objektes TOS sind die Alarmer (alarms). In FIG 3 sind die Alarmer durch eine gestrichelte Linie von den Befehlen abgetrennt. Die Darstellung gemäß FIG 3 enthält die Alarmer <alarm_1> bis <alarm_k>. Ein Technologisches Objekt hat Überwachungen und kann im Fehlerfall definierte Alarmer, gegebenenfalls mit Alarminformationen und vordefinierten Reaktionen absetzen. Die technologischen Alarmer werden am Technologischen Objekt festgestellt bzw. erzeugt. Technologische Alarmer haben eine technologieobjekttypspezifisch eingestellte Reaktion, z.B. Bewegungsstopp (die möglichen Reaktionen sind

technologieobjekttypspezifisch und daher bei den einzelnen Technologieobjekttypen explizit beschrieben). Weiterhin besitzen die technologischen Alarmer einen technologieobjekttypspezifischen Identifikator (z.B. Alarmnummer) und Parameter.

- 5 Sie besitzen somit ein einstellbares Verhalten auf die Programmbearbeitung (globale Reaktion) und erlauben weiterhin für jeden Fehler instanzspezifische Einstellungen und Reaktionen, die bei der Inbetriebnahme am Engineering-System (ES; FIG 1) vorgenommen werden.

10

Ein Anwender kann Befehle von Technologischen Objekten synchron oder asynchron nutzen, je nach Einstellung. Dadurch kann ein Befehl sowohl zyklisch geschrieben (üblich bei einer Speicherprogrammierbaren Speicherung) aber auch ereignisgesteuert (üblich bei Bewegungssteuerungen) programmiert werden.

15

Im synchronen Modus bleibt z.B. das technologische Objekt, das einen Positionierbefehl ausführt, solange in seinem Zustand, bis das Positionierziel erreicht wurde. Im asynchronen Modus dagegen läuft zeitgleich zur Ausführung des Positionierbefehls das Technologische Objekt in seinem Programmablauf weiter und kann dabei andere Zustände einnehmen. Das Technologische Objekt kann dann z.B. durch Polling geprüft werden, ob das Positionierziel erreicht wurde.

20

25

Die Darstellung gemäß FIG 4 zeigt als Verschaltungsdiagramm die Verschaltung des Technologischen Objektes "Gleichlauf" GL1 mit anderen Technologischen Objekten. Die Technologischen Objekte werden dabei als doppelt umrandete Rechtecke dargestellt, bei denen die jeweils zusammen gehörenden Ecken durch eine Verbindungslinie verbunden sind. Durch die Verschaltung des Technologischen Objektes "Gleichlauf" GL1 mit den Technologischen Objekten "Leitachse" LA1, "Folgeachse" FA1 und "Kurvenscheibe" KS1 wird ein Gleichlaufverbund hergestellt. Die Verschaltung der Technologischen Objekte erfolgt über Datenflüsse DF1 bis DF3 bzw. DF3'. FIG 4 zeigt die prinzipielle Technologieanordnung für die Realisierung eines Gleichlaufverbundes: Leitwert - Technologisches Objekt "Gleichlauf" GL1

30

35

- Technologisches Objekt "Folgeachse" FA1. In FIG 4 wird der Leitwert durch das Technologische Objekt "Leitachse" LA1 repräsentiert. Weiterhin ist in FIG 4 ist dargestellt, dass das Technologische Objekt "Leitachse" LA1 über den Datenflusspfeil DF1 den Leitwert für das Technologische Objekt "Gleichlauf" GL1 vorgibt. Das Technologische Objekt "Leitachse" LA1 kann z.B. eine Positionierachse repräsentieren. Der Leitwert kann aber auch über eine virtuelle Achse d.h. gerechnete (nicht real vorhandene) Achse oder über externe Geber für das Technologische Objekt "Gleichlauf" GL1 vorgegeben werden. Das Technologische Objekt "Gleichlauf" GL1 stellt als technologische Funktionalität Getriebegleichlauf oder Kurvengleichlauf zur Verfügung, damit können Aufsynchronisieren, Absynchronisieren oder Masterumschaltungen vorgenommen werden. Am Technologischen Objekt "Gleichlauf" GL1 kann als Gleichlaufgesetz wahlweise ein Getriebe oder eine Kurve gewählt werden. Der rechte Teil von FIG 4 stellt diese Auswahlmöglichkeiten dar. Durch den Zuordnungspfeil ZP1 ist dargestellt, dass der Schalter S1 wahlweise mit einem Getriebe, dargestellt durch den Getriebefaktor GF1 oder mit dem Technologischen Objekt "Kurvenscheibe" KS1 verbunden werden kann. Bei einer Verbindung mit dem Technologischen Objekt "Kurvenscheibe" KS1 erfolgte Datenfluss von diesem Technologischen Objekt zum Technologischen Objekt "Gleichlauf" GL1 über den Datenflusspfeil DF3, den Schalter S1 und den Datenflusspfeil DF3'. Bei einer Verbindung mit dem Getriebefaktor GF1 erfolgt der Datenfluss zum Technologischen Objekt "Gleichlauf" GL1 über den Schalter S1 und den Datenflusspfeil DF3'. Über das Technologische Objekt "Kurvenscheibe" KS1 können nicht lineare Getriebeübersetzungen am Technologischen Objekt "Gleichlauf" GL1 eingestellt werden, über den Getriebefaktor GF1 dagegen lineare Getriebeübersetzungen. Durch den Datenflusspfeil DF2 ist das Technologische Objekt "Gleichlauf" GL1 mit dem Technologischen Objekt "Folgeachse" FA1 verschaltet.

Die Darstellung gemäß FIG 4 zeigt somit die prinzipielle Konfiguration von Technologischen Objekten zur Realisierung ei-

ner Gleichlauffunktionalität und kann ihrerseits als (komplexes) Technologisches Objekt angesehen und verwendet werden.

Die Festlegung der Verschaltung der Technologischen Objekte erfolgt in der Konfigurationsphase (Projektierung). Bei Auswahlmöglichkeiten werden diese zur Laufzeit über das Anwenderprogramm (AP; FIG 2) aktiviert, d.h. zur Laufzeit können Umschaltungen programmiert werden. Prinzipiell können durch Verschaltung mehr als ein "Gleichlaufobjekt" GL1 mit einer "Folgeachse" FA1 verbunden werden, dadurch wird eine Überlagerung von Gleichlauffunktionen realisiert. Der Leitwert für das "Gleichlaufobjekt" GL1 kann auch direkt aus dem Anwenderprogramm (AP; FIG 2) vorgegeben werden. Weiterhin kann mehr als ein Technologisches Objekt für die Leitwertbereitstellung konfiguriert werden. Die aktuelle Verschaltung wird wiederum zur Laufzeit über Befehle im Anwenderprogramm (AP; FIG 2) ausgewählt und aktiviert. Außerdem kann für die Festlegung des Gleichlaufgesetzes zwischen verschiedenen Technologischen Objekten "Kurvenscheibe" KS1 und/oder zwischen verschiedenen Getriebefaktoren GF1 durch Programmierung online umgeschaltet werden. Ein Technologisches Objekt "Kurvenscheibe" KS1 kann einem oder mehreren Technologischen Objekten "Gleichlauf" GL1 zugeordnet werden. Weiterhin können von einem Technologischen Objekt "Leitachse" LA1 eine oder mehrere Gleichlaufverbindungen über Technologische Objekte "Gleichlauf" GL1 konfiguriert werden.

Darstellung gemäß FIG 5 zeigt einen Gleichlaufverbund mit Umschaltmöglichkeiten zwischen verschiedenen Leitwertquellen und Gleichlaufgesetzen, ebenfalls in Form eines Verschaltungsdiagramms. In FIG 5 kann das Technologische Objekt "Gleichlauf" GL2 Leitwerte von den Technologischen Objekten "Zeit" T, "virtuelle Achse" VA1, "Leitachse" LA2, "Leitachse" LA3, "externer Geber" EG1 sowie von einem Programmwert TV des Anwenderprogrammes (AP; FIG 2) bekommen. Durch den Zuordnungspfeil ZP2 ist angedeutet, dass der Schalter S2 unterschiedliche Leitwertverbindungen für das Technologische Ob-

jekt "Gleichlauf" GL2 herstellen kann. Über einen der Datenflüsse DF4 bis DF8 sowie über den Schalter S2 und den Datenfluss DF12 wird die „Leitwertverschaltung“ zum Technologischen Objekt "Gleichlauf" GL2 erreicht. Die Technologischen

5 Objekte "Zeit" T, "virtuelle Achse" VA1, "Leitachse" LA2 und LA3, "externer Geber" EG1 sowie der Programmwert TV sind die potentiellen Master für das Technologische Objekt "Gleichlauf" GL2. Die möglichen Verschaltungen werden projiziert und die Auswahl eines projizierten Masters kann zur Laufzeit

10 aus dem Anwenderprogramm (AP; FIG 2) erfolgen. Damit sind Masterumschaltungen möglich. Das Technologische Objekt "virtuelle Achse" VA1 repräsentiert nicht eine real vorhandene Achse sondern eine gerechnete Achse. "Virtuelle Achsen" sind dadurch gekennzeichnet, dass sie über Befehle kommandiert

15 werden können und eine Bewegungsführung bzw. Interpretation besitzen, aber keine Regelung und keinen Antrieb. Die Technologischen Objekte "Leitachse" LA2 und LA3 repräsentieren dagegen reale Achsen. Reale Achsen repräsentieren Standardachsen mit Antrieb, Motor, Geber, sie besitzen also einen realen

20 Aktor. Auch das Technologische Objekt "externer Geber" EG1 kann einen Leitwert für das Technologische Objekt "Gleichlauf" GL2 bereitstellen. Ein "externer Geber" EG1 besitzt üblicherweise keine Achse und stellt die Informationen in einem projektierbaren Format bereit. "Externe Geber" sind z.B. Winkelgeber an einer Presse. Auch vom Technologischen Objekt

25 "Zeit" T und vom Programmwert TV können Leitwerte für das Technologische Objekt GL2 bereitgestellt werden. Ein Technologisches Objekt "Zeit" stellt einen Leitwert in Form eines Zeitwertes bzw. Zeitfaktors bereit, die Projektierung eines

30 Programmwertes DV als Leitwert erfolgt im Anwenderprogramm (AP; FIG 2). Die Technologischen Objekte sind hierbei in der üblichen Notation dargestellt.

In FIG 5 ist dargestellt, dass als Gleichlaufgesetz für das

35 Technologische Objekt "Gleichlauf" GL2 wahlweise ein Getriebefaktor GF2 oder die Technologischen Objekte "Kurvenscheibe" KS2 und KS3 gewählt werden können. Durch den Zuordnungspfeil

ZP3 ist dargestellt, dass der Schalter S3 wahlweise zwischen den Technologischen Objekten KS2, KS3 und dem Getriebefaktor GF2 eingestellt werden kann. Die "getriebemäßige Verschaltung" mit dem Technologischen Objekt "Gleichlauf" GL2 erfolgt dann über die Datenflusspfeile DF9, DF10, den eingestellten Schalter S3 sowie über den Datenflusspfeil DF11. Die Schalterverbindungen S2 und S3 sind im Anwenderprogramm (AP; FIG 2) programmierbar. Über den Datenflusspfeil DF 13 ist das Technologische Objekt "Gleichlauf" GL2 mit dem Technologischen Objekt "Folgeachse" FA2 verbunden. Das Technologische Objekt "Gleichlauf" GL2 wird bei der Projektierung also slaveseitig mit dem Technologischen Objekt "Folgeachse" FA2 das z.B. eine Gleichlaufachse repräsentieren kann, verschaltet. Masterseitig wird das Technologische Objekt "Gleichlauf" GL2 mit einem Technologischen Objekt verschaltet, dass einen Leitwert zur Verfügung stellt, dieser Leitwert kann auch direkt aus dem Anwenderprogramm (AP; FIG 2) vorgegeben werden. Somit kann mehr als ein Technologisches Objekt für die Leitwertbereitstellung konfiguriert werden, die aktuelle Verschaltung wird zur Laufzeit über Befehle im Anwenderprogramm ausgewählt.

Die Darstellung gemäß FIG 6 zeigt die Verschaltung des Technologischen Objektes "Messtaster" MT1. Die Technologischen Objekte sind hierbei in der üblichen Notation dargestellt. Das Technologische Objekt "Messtaster" MT1 stellt die Funktionalität zur Durchführung eines Messauftrages bereit. Für die Funktionen am Technologischen Objekt "Messtaster" MT1 können Messaufträge aktiviert und parametrisiert werden. Über den Messeingang ME und den Datenflusspfeil DF14 wird der Messwert an das Technologische Objekt "Messtaster" MT1 geliefert. Der Messeingang ME ist als Ellipse dargestellt. Ein Messeingang ME kann mit mehreren Technologischen Objekten "Messtaster" verschaltet sein. Diese Technologischen Objekte "Messtaster" können dabei auch gleichzeitig aktiviert sein. Ein Messeingang ME entspricht dabei üblicherweise einem Hardware-Messeingang der dem Technologischen Objekt "Messtaster" MT1 über Konfiguration zugeordnet wird. Weiterhin ist das Techno-

logische Objekt "Messtaster" MT1 mit mindestens einem Technologischen Objekt verschaltet, das einen Messwert (z.B. Position) liefert. In FIG 7 ist das Technologische Objekt "Messtaster" MT 1 mit den Technologischen Objekten "Achse" A1 und "externer Geber" EG2 über die Datenflusspfeile DF15 bzw. DF16 verschaltet. Das Technologische Objekt "Achse" A1 kann z.B. eine Positionierachse oder eine Gleichlaufachse sein. Ein Technologisches Objekt, das einen Messwert liefert, kann mit mehreren Technologischen Objekten "Messtaster" verschaltet werden.

Die Darstellung gemäß FIG 7 zeigt in einem Verschaltungsdiagramm das Technologische Objekt "Nocke" N1, verschaltet mit den Technologischen Objekten "Achse" A2 und "externer Geber" EG3. Das Technologische Objekt "Achse" A2 ist über den Datenflusspfeil DF17, das Technologische Objekt "externer Geber" EG3 ist über den Datenflusspfeil DF18 mit dem Technologischen Objekt "Nocke" N1 verschaltet. Über den Datenflusspfeil DF19 ist das Technologische Objekt "Nocke" N1 mit dem Ausgang Out verschaltet, der Ausgang Out ist als Ellipse dargestellt. Das Technologische Objekt "Nocke" N1 stellt die Funktionalität zur Berechnung von Nockenschaltwerten bereit. Über die Funktionen am Technologischen Objekt "Nocke" N1 können Nockenfunktionen aktiviert und parametrisiert werden. Die Technologischen Objekte "Achse" A2 bzw. "externer Geber" EG3 stellen die Bezugswerte für das Technologische Objekt "Nocke" N1 bereit. Die Zuordnung dieser Technologischen Objekte zum Technologischen Objekt "Nocke" N1 wird vom Anwender projiziert. Der Anwender projiziert weiterhin die Zuordnung des Technologischen Objektes "Nocke" N1 zu einem Ausgang Out, dabei ist auch eine Zuordnung auf interne Variablen möglich. Für eine aktuelle Anwendung ist das Technologische Objekt "Nocke" N1 mit genau einem Technologischen Objekt verschaltet, das den Bezugswert liefert.

35

Ein Bezugswert ist z.B. ist eine Achsposition. Hierbei kann das Technologische Objekt "Achse" A2 z.B. eine Positionier-

achse oder eine Gleichlaufachse repräsentieren. Dies ist möglich, dass eine Zuordnung des Technologischen Objektes "Nocke" N1 auf einen Ausgang Out entfallen kann, dann wirkt das Technologische Objekt "Nocke" N1 nur auf Systemvariablen am Technologischen Objekt (z.B. für die Verwendung des Technologischen Objektes als interne Nocke). Das Technologische Objekt, das den Bezugswert liefert, kann mit mehreren auch unterschiedlichen Technologischen Objekten Nocken gleichzeitig verschaltet sein. Die Technologischen Objekte sind hierbei in der üblichen Notation dargestellt.

In der Darstellung gemäß FIG 8 wird gezeigt, dass ein Technologisches Objekt "Folgeachse" FA3 mit mehreren Technologischen Objekten "Gleichlauf" GL3 und GL4 verschaltet sein kann. Das Technologische Objekt "Folgeachse" FA3 ist durch den Datenflusspfeil DF22 mit dem Technologischen Objekt "Gleichlauf" GL3 und mit dem Datenflusspfeil DF23 mit dem Technologischen Objekt "Gleichlauf" GL4 verschaltet. Die Technologischen "Gleichlaufobjekte" GL3 und GL4 erhalten über die Datenflusspfeile DF20 bzw. DF21 ihre Leitwertvorgaben. In FIG 8 ist dargestellt, dass die Leitwerte für den jeweiligen Gleichlaufverbund über unterschiedliche Technologische Objekte erfolgen kann. So können für das Gleichlaufobjekt GL3 z.B. das Technologische Objekt "Achse" A3, das Technologische Objekt "virtuelle Achse" VA2 oder das Technologische Objekt "externer Geber" EG4 den Leitwert bereitstellen. Für das "Gleichlaufobjekt" GL4 kann dementsprechend der Leitwert z.B. von den Technologischen Objekten "Achse" A4, "virtuelle Achse" VA3 oder "externer Geber" EG5 bereitgestellt werden. In FIG 8 bilden dann z.B. die Technologischen Objekte "Achse" A4, "Gleichlauf" GL4 und "Folgeachse" FA3 einen Gleichlaufverband. Die jeweils gewünschte Verschaltung wird vom Anwender projiziert, die Auswahl eines projizierten Masters (der Master stellt den Leitwert für den Gleichlaufverbund zur Verfügung) kann zur Laufzeit aus dem Anwenderprogramm erfolgen, damit sind Masterumschaltungen möglich. In FIG 8 stellt das Technologische Objekt "Folgeachse" FA3 den Slave im Gleich-

laufverbund dar. Die Technologischen Objekte sind hierbei in der üblichen Notation dargestellt.

Darstellung gemäß FIG 9 zeigt ein Verschaltungsdiagramm bei dem das Technologische Objekt "Kurvenscheibe" KS3 das Getriebe-
5 begesetz für zwei "Gleichlaufobjekte" GL5 und GL6 über die Datenflusspfeile DF26 bzw. DF27 bereitstellt. In FIG 9 sind somit zwei Gleichlaufverbunde dargestellt, die jeweils vom gleichen Technologischen Objekt "Kurvenscheibe" KS2 mit einem
10 gemeinsamen Getriebe-
Gegensatz versorgt werden. Die beiden Gleichlaufverbunde sind links und rechts vom Technologischen Objekt "Kurvenscheibe" KS3 angeordnet. Der linke Gleichlaufverbund wird gebildet durch das Technologische Objekt "Achse" A5, das den Leitwert bereitstellt und somit als Leitachse
15 gilt. Es kann sich dabei z.B. um eine Positionier- oder Gleichlaufachse handeln. Das Technologische Objekt "Achse" A5 ist mit dem Datenflusspfeil DF24 mit dem "Gleichlaufobjekt" GL5 verbunden. Über diesen Datenflusspfeil DF24 wird der Leitwert bereitgestellt. Auf der Slave-seite ist das Technologische Objekt "Gleichlauf" GL5 über den Datenflusspfeil GF25
20 mit dem Technologischen Objekt "Folgeachse" FA4 verbunden. Der rechte Gleichlaufverbund wird gebildet durch die Technologischen Objekte "Achse" A6, "Gleichlauf" GL6 und "Folgeachse" FA5. Die "Achse" A6 entspricht dabei der Leitachse, die "Folgeachse" FA5 repräsentiert eine Slaveachse. Die Verschaltung erfolgt hierbei über die Datenflusspfeile DF28 bzw. DF29. Weiterhin ist es möglich, das von einer Leitachse aus eine oder mehrere Gleichlaufverbindungen über Gleichlaufobjekte konfiguriert werden. Technologische Objekte "Kurven-
30 scheibe" können einem oder mehreren Gleichlaufobjekten zugeordnet werden. Die Zusammenstellung von Gleichlaufverbund wird vom Anwender projektiert. Projektierte Gleichlaufverbunde können wiederum als Technologische Objekte repräsentiert werden und ihre Funktionalität in anderen Applikationen wieder verwendet werden. Die Technologischen Objekte sind hier-
35 bei in der üblichen Notation dargestellt.

Die Darstellung gemäß FIG 10 zeigt die Zusammenfassung von mehreren Technologischen Objekten zu einem Technologiepaket TP. Das Technologiepaket TP ist dabei als Rechteck dargestellt, wobei die linke obere Ecke abgeschnitten ist. Das

5 Technologiepaket TP enthält die Technologischen Objekte "Nocke" N2, "externer Geber" EG6, "Drehzahlachse" DrehA, "Mess-taster" MT2 sowie "Positionierachse" PosA. Die Technologischen Objekte sind dabei in der üblichen Notation dargestellt. Die Technologischen Objekte repräsentieren dabei kei-

10 ne Instanzen, sondern Technologieobjekttypen. Ein Technologiepaket TP enthält somit eine Ansammlung von Technologieobjekttypen, die gewisse Funktionalitäten repräsentieren. Die Zuladung von Technologischen Objekten ins Runtime-System der Steuerung und damit die funktionelle Erweiterung der Steue-

15 rung erfolgt über Technologiepakete. Ein Anwender kann sich bestimmte Technologiepakete TP die wiederum Technologieobjekttypen enthalten ins Runtime-System (RTS4, RTS5; FIG 2) laden und somit eine technologische Skalierung der Funktionalität der Steuerung erreichen. Weiterhin kann durch die Tech-

20 nologiepakete TP bei entsprechender Zuordnung von Technologieobjekttypen eine funktionelle Strukturierung erreicht werden.

Patentansprüche

1. Industrielle Steuerung für technische Prozesse (P), insbesondere für Produktionsmaschinen,

5 d a d u r c h g e k e n n z e i c h n e t,
dass

10 a) die Steuerung ein allgemein einsetzbares, vorzugsweise
technologieneutrales Basissystem (UMC-K) für die Steuerungsgrundfunktionalität besitzt,

15 b) die technologische Funktionalität der Steuerung über
Technologische Objekttypen (TO1 - TOn) bereitgestellt
ist,

c) eine technologische Skalierung hinsichtlich der Funktionalität der Steuerung durch zuladbare Technologische Objekttypen (TO1 - TOn) erfolgt und

20 d) eine auf die jeweilige Applikation vom Anwender
zuschneidbare Auswahl und Instanziierung der Objekttypen
erfolgt.

2. Industrielle Steuerung nach Anspruch 1,

d a d u r c h g e k e n n z e i c h n e t,

dass das Laden und die Verteilung der Technologischen Objekte (TO1 - TOn) auf unterschiedlich oder gleich performante Hardware-Systeme und/oder Laufzeitsysteme erfolgt.

30 3. Industrielle Steuerung nach Anspruch 2,

d a d u r c h g e k e n n z e i c h n e t,

dass das Laden und die Verteilung der Technologischen Objekte (TO1 - TOn) auf unterschiedlich oder gleich performante Hardware-Systeme und/oder Laufzeitsysteme innerhalb eines Projektes erfolgt, wobei sich ein Projekt auf Daten und/oder Programme von einer oder mehreren Steuerungseinheiten bezieht.
35

4. Industrielle Steuerung nach einem der vorstehenden Ansprüche,

d a d u r c h g e k e n n z e i c h n e t,

dass die Verteilung der Funktionalität der Technologischen

5 Objekte (TO1 - TOn) auf miteinander in Echtzeit durch takt-synchron äquidistant kommunizierende Steuerungseinheiten erfolgt.

10 5. Industrielle Steuerung nach einem der vorstehenden Ansprüche,

d a d u r c h g e k e n n z e i c h n e t,

dass die Funktionalität der Steuerung durch die Zuladbarkeit beliebiger Technologischer Objekte (TO1 - Ton) flexibel erweiterbar ist.

15

6. Industrielle Steuerung nach einem der vorstehenden Ansprüche,

d a d u r c h g e k e n n z e i c h n e t,

20 dass die Technologischen Objekte (TO1 - Ton) Programmierinterfaces und/oder Systemvariablen und/oder Alarmerhalten.

7. Industrielle Steuerung nach einem der vorstehenden Ansprüche,

d a d u r c h g e k e n n z e i c h n e t,

25 dass die Technologischen Objekte (TO1 - Ton) in eine Programmierungsumgebung integriert sind.

8. Industrielle Steuerung nach Anspruch 7,

d a d u r c h g e k e n n z e i c h n e t,

30 dass eine Standardprogrammierungsumgebung verwendet wird.

9. Industrielle Steuerung nach einem der vorstehenden Ansprüche,

d a d u r c h g e k e n n z e i c h n e t,

35 dass eine rückwirkungsfreie Programmierung eines Technologischen Objektes (TO1 - Ton) bezüglich der anderen vorhandenen Technologischen Objekte (TO1 - Ton) und des Steuerungsbasis-

systems, sofern nicht explizit eine Rückwirkung programmiert bzw. projiziert ist, vorgesehen ist.

5 10. Industrielle Steuerung nach einem der vorstehenden Ansprüche,

d a d u r c h g e k e n n z e i c h n e t,
dass die Nutzung der Technologischen Objekte (TO1 - Ton)
Hardware- bzw. Plattformneutral erfolgt.

10 11. Industrielle Steuerung nach einem der vorstehenden Ansprüche,

d a d u r c h g e k e n n z e i c h n e t,
dass Technologische Objekte (TO1 - TOn) bezüglich ihrer Funk-
tionalität zu anderen Technologischen Objekten (TO1 - TOn)
15 hierarchische Beziehungen und/oder Datenflussbeziehungen (DF1
- DF29, DF3') aufweisen.

12. Industrielle Steuerung nach einem der vorstehenden Ansprüche,

20 d a d u r c h g e k e n n z e i c h n e t,
dass insbesondere für Bewegungssteuerungen Elemente wie Ach-
sen (A1 - A6), Geber (EG1 - EG6), Nocken (N1, N2), Messtaster
(MT1, MT2), Gleichlauf (GL1 - GL6) und Kurvenscheiben (KS1 -
KS3) als Technologische Objekte (TO1 - TOn) vorhanden sind.

25 13. Industrielle Steuerung nach einem der vorstehenden Ansprüche,

d a d u r c h g e k e n n z e i c h n e t,
dass die Technologieobjekttypen zu Technologiepaketen (TP)
30 zusammenfassbar sind.

14. Industrielle Steuerung nach einem der vorstehenden Ansprüche,

35 d a d u r c h g e k e n n z e i c h n e t,
dass die Programmerstellung vor Zuordnung der Hardware-
Plattform erfolgt.

15. Industrielle Steuerung nach einem der vorstehenden Ansprüche,

d a d u r c h g e k e n n z e i c h n e t,

5 dass die Anzahl der Instanziierungen von Technologieobjekttypen flexibel einstellbar ist.

16. Industrielle Steuerung nach einem der vorstehenden Ansprüche,

d a d u r c h g e k e n n z e i c h n e t,

10 dass die Instanzen von Technologieobjekttypen auf einer oder auf mehreren Steuerungseinheiten verteilt abgelegt werden.

17. Industrielle Steuerung nach einem der vorstehenden Ansprüche,

15 d a d u r c h g e k e n n z e i c h n e t,

dass die Befehle von Technologischen Objekten (TO1 - TOn) synchron und/oder asynchron genutzt werden.

18. Verfahren zur Erstellung einer industrielle Steuerung,

20 insbesondere für ein bestimmtes Einsatzgebiet,

g e k e n n z e i c h n e t d u r c h die aufeinander folgenden Schritte:

25 a) Verwendung eines Basissystems (UMC-K) mit einer vorzugsweise technologieneutralen Grundfunktionalität,

30 b) Auswahl und Instanziierung der für die Anwendung relevanten Technologischen Objekttypen (TO1 - TOn) und Konfiguration der erhaltenen Technologischen Objekte (TO1 - TOn) im Engineering-System (ES),

c) Hinzuladen der ausgewählten Technologischen Objekttypen (TO1 - TOn) ins Runtime-System (RTS1 - RTS5),

35 d) Laden und Nutzung der instanziierten Technologischen Objekte und der Instanziierungsinformation im Runtime-System (RTS1 - RTS5),

- e) Erstellung der Anwenderprogramme (AP) unter Nutzung der
instanziierten Technologischen Objekte (TO1 - TOn) und
 - f) Laden der Anwenderprogramme (AP) ins Runtime-System
(RTS1 - RTS5).
- 5

Zusammenfassung

Industrielle Steuerung auf der Basis Technologischer Objekte

- 5 Durch die Zuladbarkeit von Technologischen Objekttypen (TO1-
TOn) in das Runtime-System (RTS1 - RTS5) einer industriellen
Steuerung wird das Basissystem (UMC-K) der Steuerung funktio-
nell erweitert und eine technologische Skalierung der Steue-
10 rung erreicht. Die zugeladenen Technologischen Objekttypen
sind beliebig instanziiierbar. Das Zuladen erfolgt in Form von
Technologiepaketen (TP). Der Anwender kann in seinen Anwen-
derprogrammen (AP) diese Funktionalität direkt verwenden.

15 FIG 2

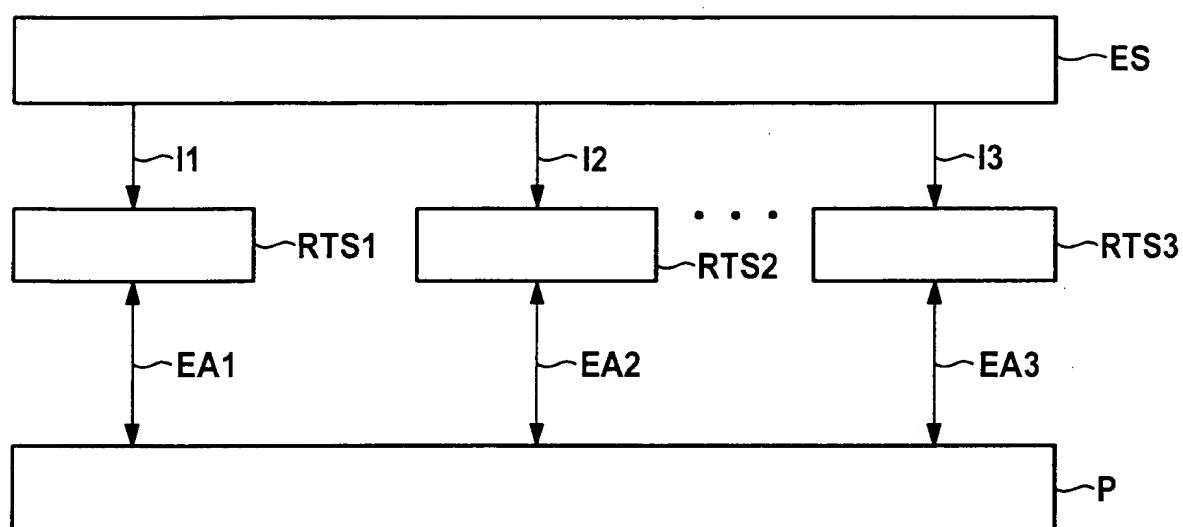


FIG 1

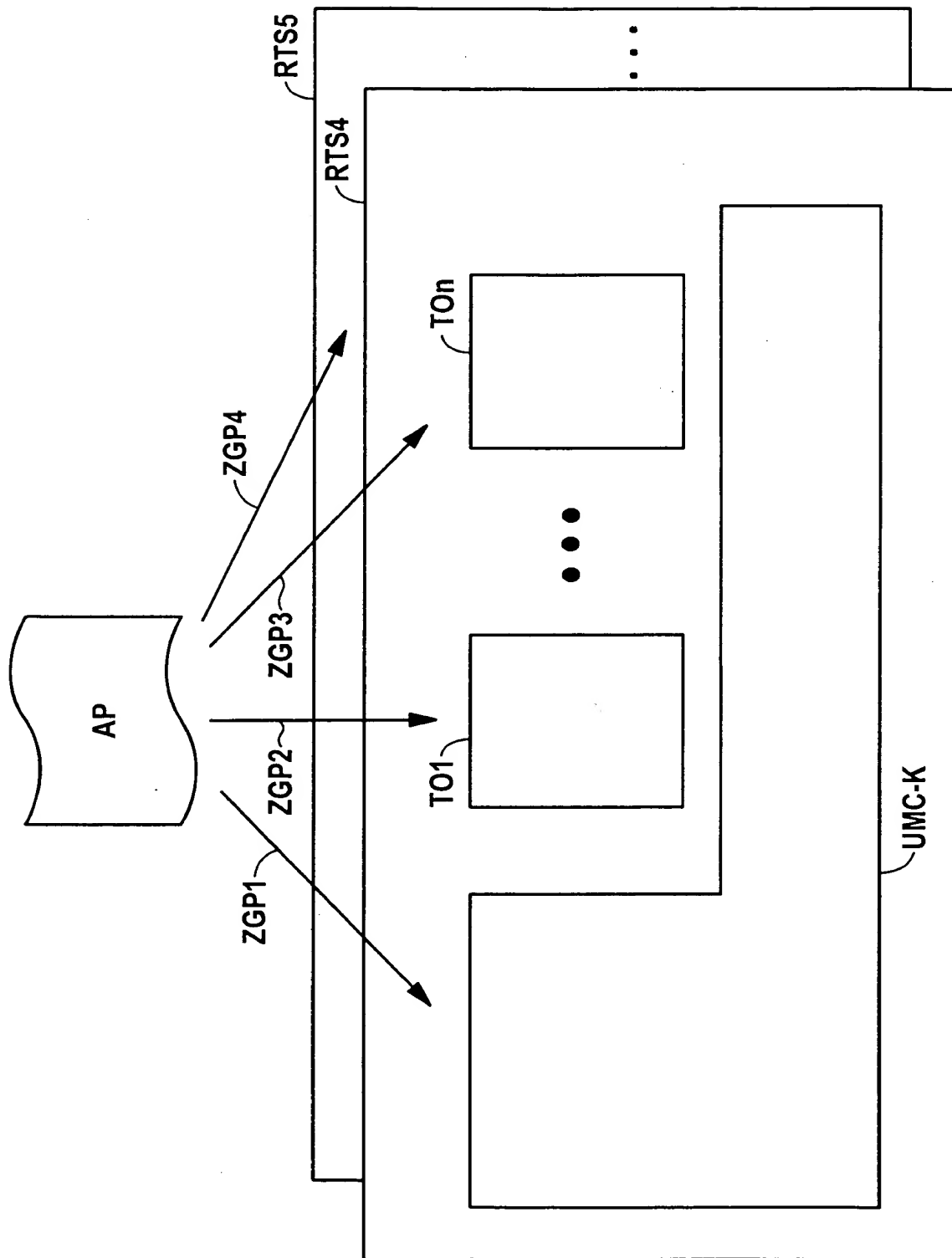


FIG 2

3/10

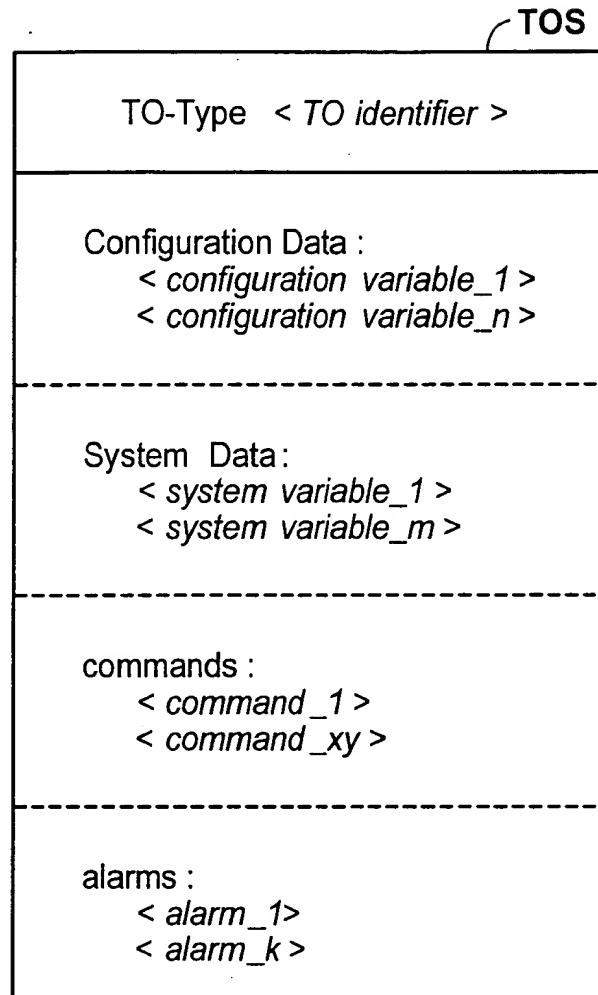


FIG 3

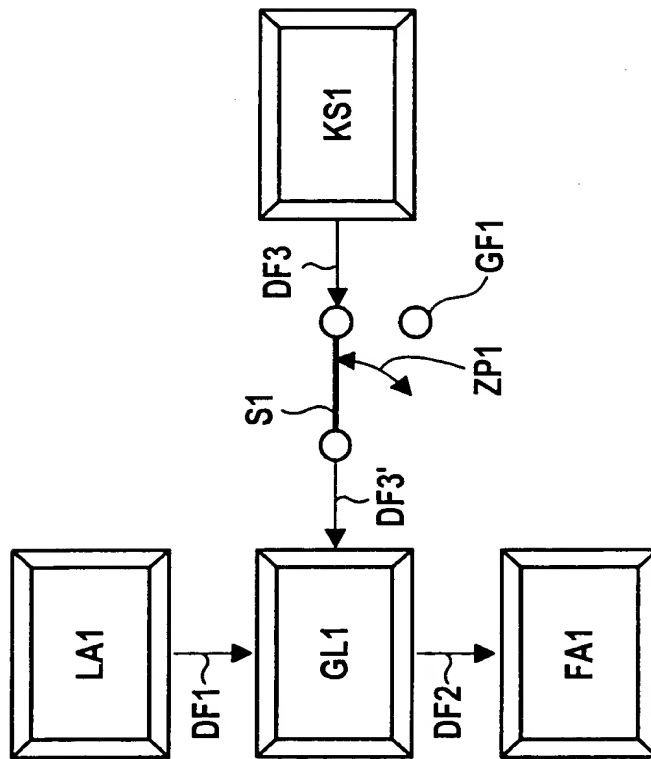


FIG 4

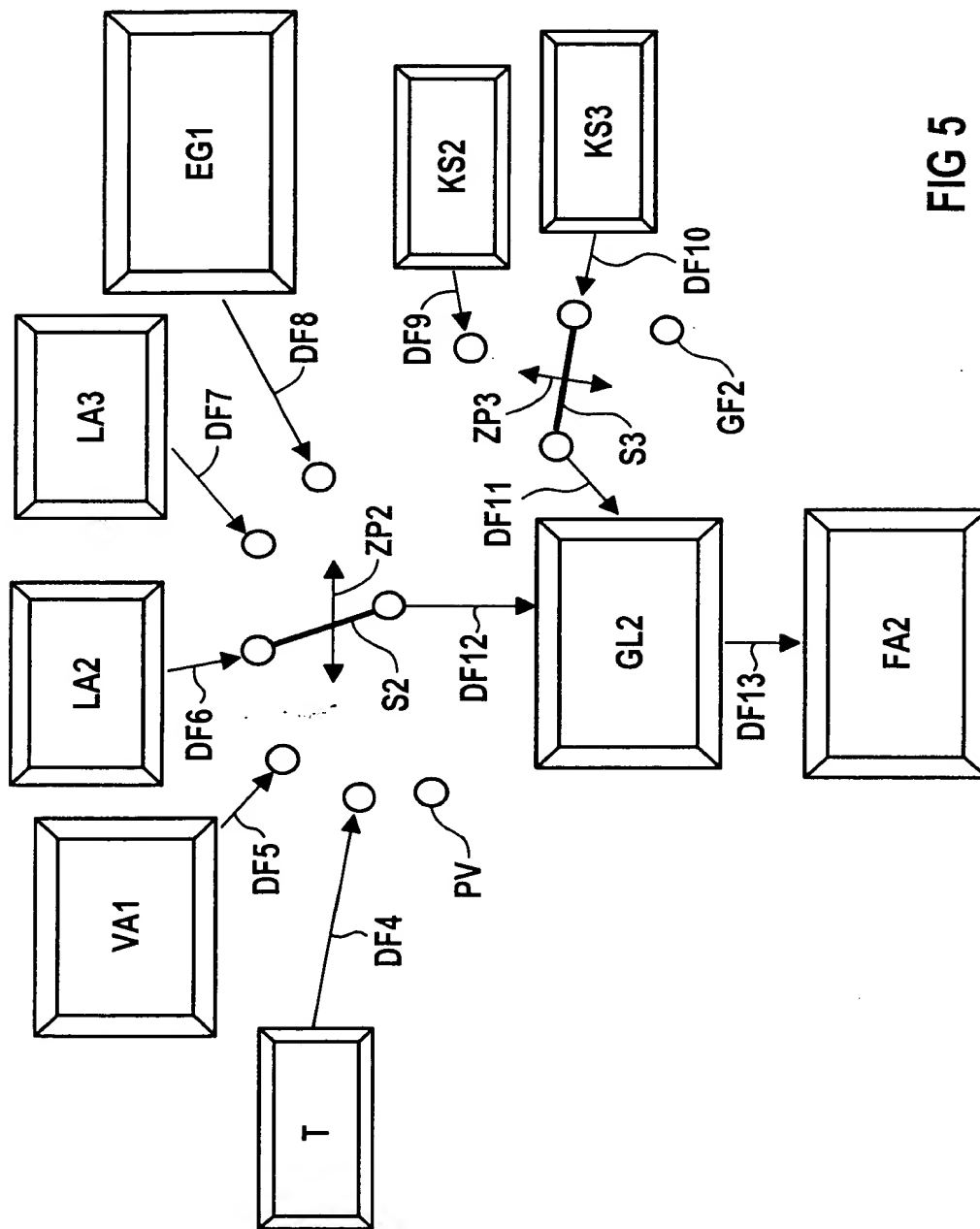


FIG 5

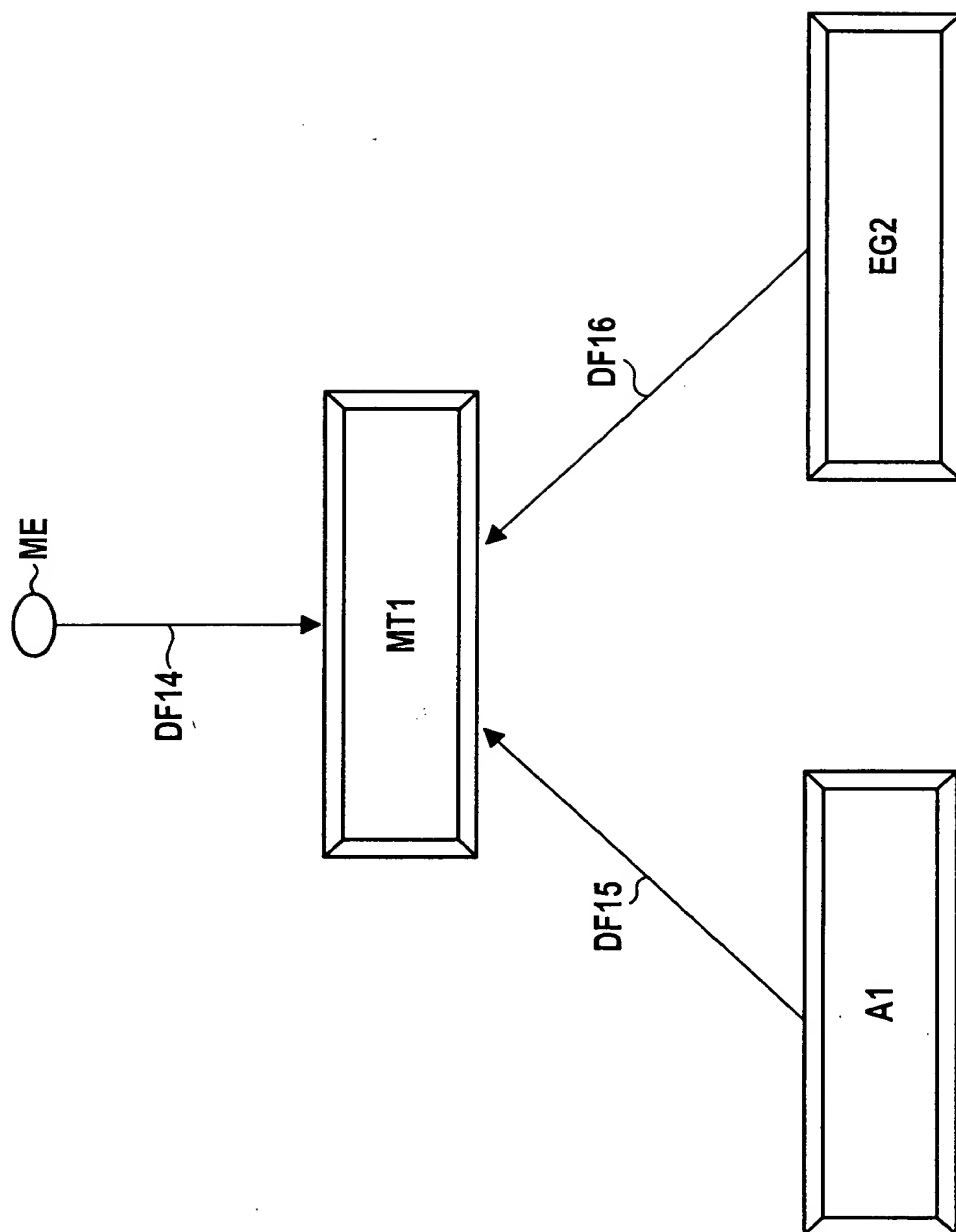
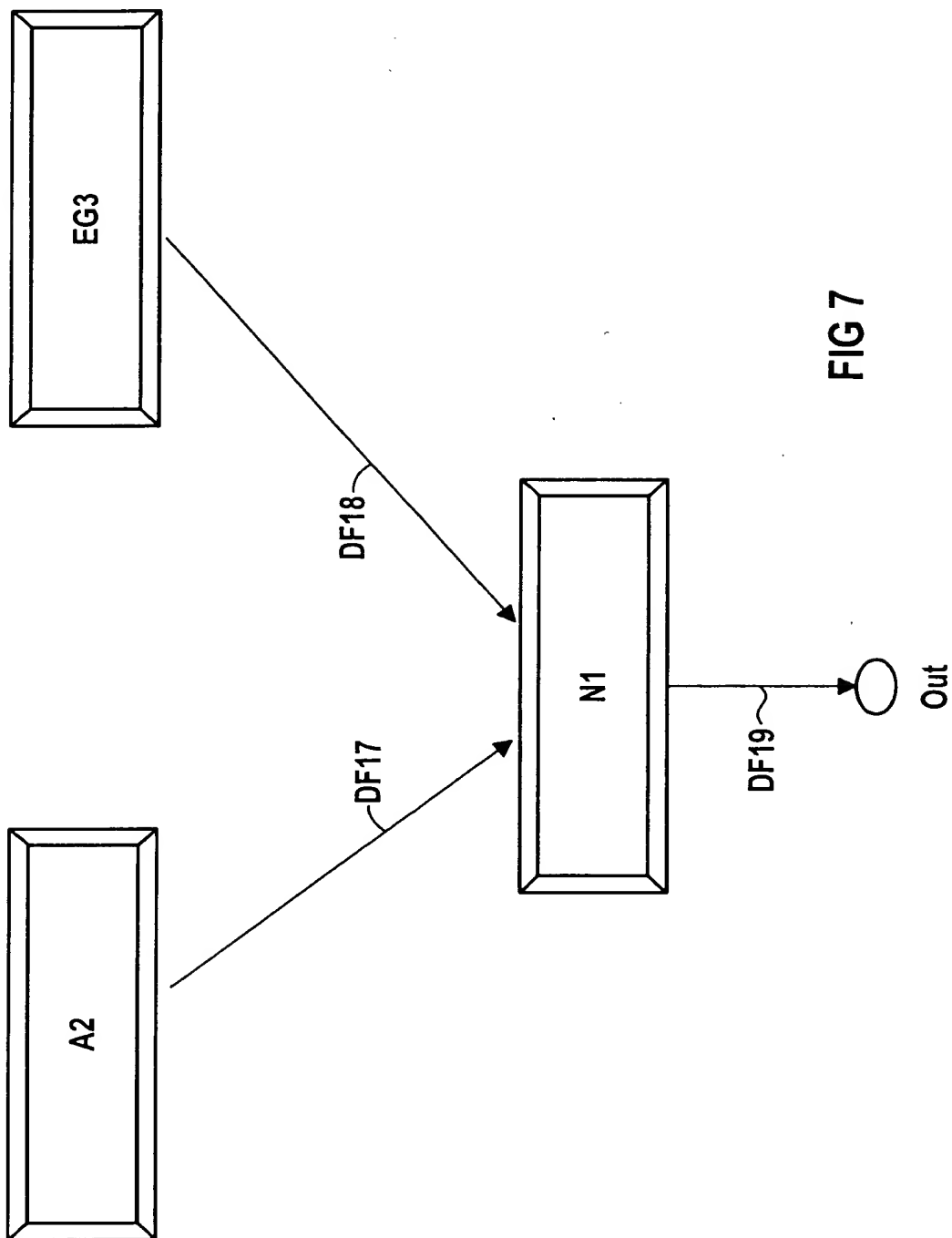
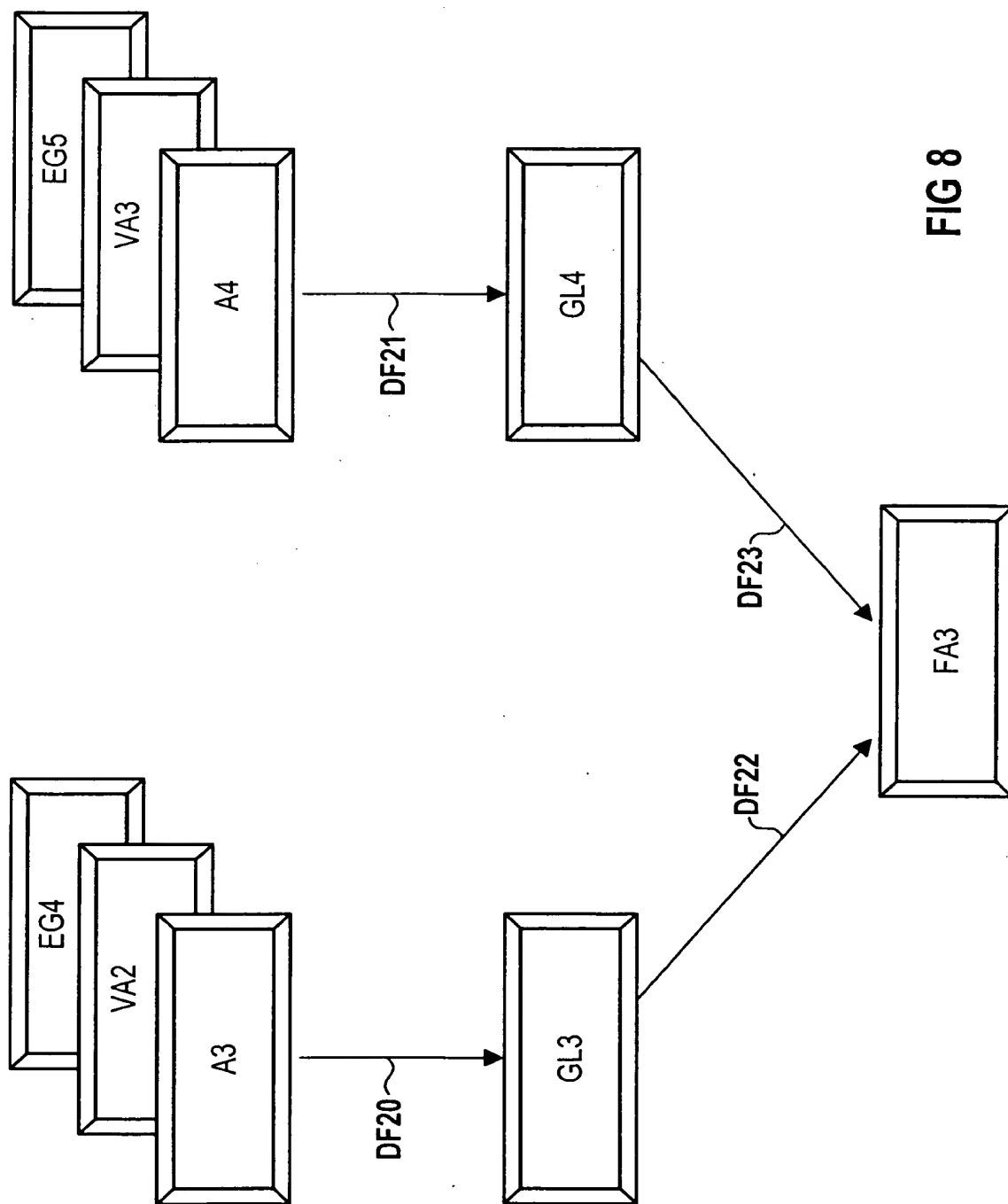


FIG 6

7/10





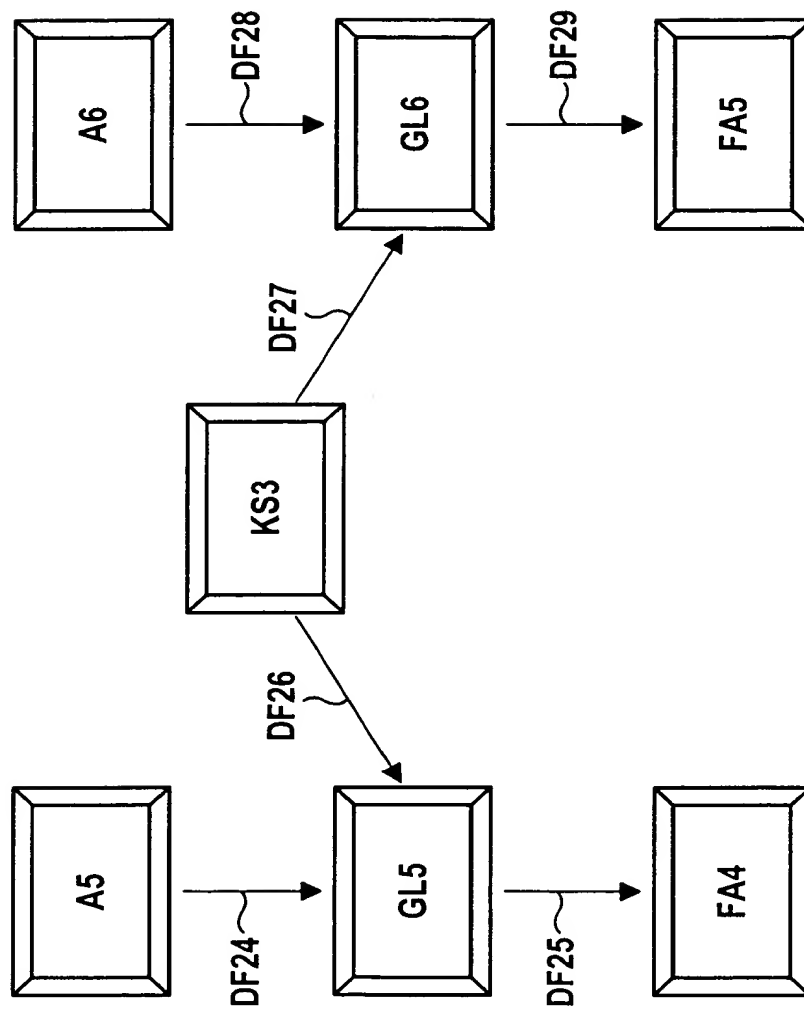


FIG 9

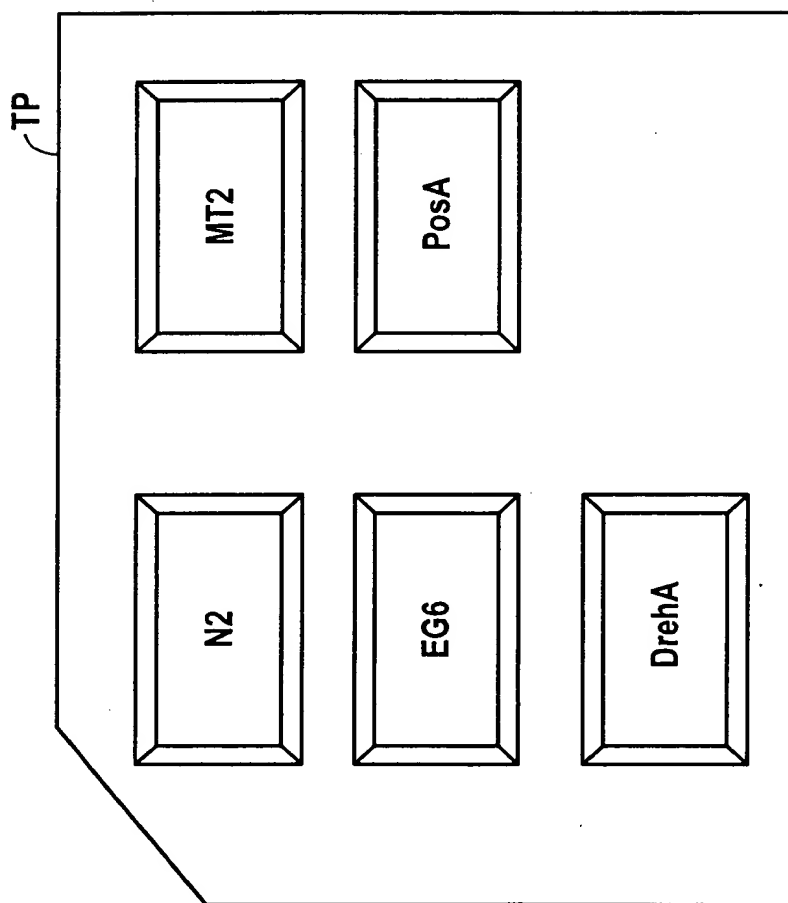


FIG 10